

# 9接口、命令

2019年2月1日 16:52



## ◆ 用户接口

### 一. 字符显示式联机用户接口/联机命令接口

- 1) 指在终端键入命令，终端处理程序接收，终端屏幕反馈
  - 2) 命令：由动词和参数组成，有规定词法、语法、语义和表达形式
  - 3) 命令语言：命令为基本单位，指示操作系统完成特定功能
  - 4) 命令集：诸多命令的集合，完整的命令集包含操作系统提供的全部功能
1. 命令行方式：以行为单位，一般每行不超过256字符，回车符为结束标记
    - 1) 间断式的串行执行方式：后一个命令的输入需等待前一命令执行结束
    - 2) 并行执行方式：在命令结尾输入特定标记，将其作为后台命令处理
    - 3) 简单命令的形式：Command arg1 arg2.....
      - (1) Command是命令名/命令动词，其余的为执行参数，有时可缺省
  2. 批命令方式：将一系列命令组织在文件中，一次建立，多次执行
    - 1) 如MS-DOS后缀为.bat的文件就是批命令文件
    - 2) 节省了时间，减少了出错率，方便了用户
    - 3) 一般操作系统会提供子命令和形式参数书写批命令文件
    - 4) 如UNIX和Linux的Shell不仅是交互型命令解释程序，还有一种命令级程序设计语言解释系统，允许用户使用Shell简单命令、位置参数和控制流语句编制带形参的批命令文件，称为Shell文件或Shell过程，Shell可自动解释和执行该文件/过程中的命令

### 二. 图形化联机用户接口Graphics User Interface

1. 其引入：1981年Xerox首次在Star8010推出，1983年苹果也在Lisa和Macintosh上成功使用，之后微软的windows、ibm的os/2、UNIX和Linux的X-Window也使用了GUI。现已有国际GUI标准
2. WIMP技术：Window窗、Icon图标、Menu菜单、Pointing device鼠标
3. Windows 的GUI：系统初始化后，OS为终端用户生产了一个运行explorer.exe的进程，运行一个具有窗口界面的命令解释程序，即桌面，之后点击某程序对应的图标，即会弹出新进程对应的窗口
  - 1) 是事件驱动控制方式，用户的动作产生事件，驱动程序工作
  - 2) 由中断系统引出事件驱动控制程序，接收、分析、处理、清除事件
4. 联机命令接口可对资源进行更多更深入的控制，仍受高级用户和程序员的欢迎

### 三. 联机命令的类型

1. 系统访问类
  - 1) Login键入管理员处获得的注册名
  - 2) Password键入密码，一般此时系统会关闭回显，且多次输错会解除连接
2. 文件操作
  - 1) 显示文件：type、拷贝文件：copy、比较文件：comp
  - 2) 重命名：Rename

3) 删除: erase。例: 参数为\*.BAK时删除指定目录所有扩展名为BAK的文件

### 3. 目录操作

1) 建立子目录: mkdir

2) 显示目录项: dir

3) 删除空子目录: rmdir (只包含.和..)

4) 显示目录结构: tree

5) 改变当前目录: chdir。例: 参数为..时会返回到上级目录

### 4. 其他

1) 输出重定向: > 文件或设备、输入重定向: 文件或设备 <

2) 管道连接: 命令 | 命令

3) 过滤。如MS-DOS的find/N输出含有指定字串的行, find/C输出其行数, find/V输出不含指定字串的行数

4) 批命令。如MS-DOS中用batch命令执行批命令文件

◆

#### ◆ Shell命令语言

### 1. Linux的Shell是命令语言、命令解释器程序、程序设计语言的统称

1) 作为命令语言, 有自己内建的命令集, 向用户提供操作系统的接口

2) 作为设计语言, 支持函数、变量、数组、程序控制结构

3) 作为命令解释器程序, 可对输入的命令解释执行

### 一. 简单命令简介

1) 简单命令: 一条命令行仅有一个命令

2) 一条简单命令便是一个目标程序的名字

### 1. 简单命令的格式

1) UNIX和Linux都规定用小写字母

(1) 选项: -开始的, 后跟多个字母或数字的可选自变量

(2) \$: Linux的默认系统提示符

(3) 例: \$ ls -tr file1 file2按最近修改序和反字母序打印两目录的目录项

### 2. 简单命令的分类

1) 可分为系统提供的标准命令: 调用各种语言处理、实用程序等, 管理员可增添, 和用户自定义的命令

2) 也可按是否常驻内存分为内部命令、外部命令。如改变工作目录cd是内部的, 拷贝cp, 移动rm是外存某目录上的

3) 简单命令的管理是对用户透明的, 搜索路径能找到对应应用程序的即可视作系统调用, 由Linux内核处理

### 3. Shell的种类

1) Bourne Shell, 简称B Shell。是UNIX最初使用的Shell。比C Shell小, 效率高, 交互性略差

2) Bourne Again Shell简称Bash。是B Shell的一个版本, 扩展了命令补全、命令编辑、命令历史表等功能。灵活强大友好, 是Linux默认的

3) C Shell, 简称C Sh。更适合编程, 是标准Berkeley System Distribution

命令解释。语法类似C。提示符为%。兼容B Shell，提供更多特殊功能，如!  
!表示重复执行，!!表示重复执行最后输入的命令

- 4) Tcsh，是C Shell的一个扩展版本，包括命令行编辑、可编程单词补全、拼写矫正、历史命令替换、作业控制等，是Linux的默认C Shell
- 5) Korn Shell，简称K Sh，结合了B Shell和C Sh的有点，和B Shell兼容
- 6) Pdksh，是Linux上的K Sh扩展，支持在命令行上挂起、后台执行、唤醒、终止程序

## 二. 简单命令的类型

### 1. 进入/退出系统Login/Logout

- 1) 进入系统/注册，管理员用用户名，在系统文件树建立子目录树根结点
- 2) 退出系统，由系统为用户记账，清除用户使用环境，一般是按ctrl d

### 2. 文件操作：显示内容cat、复制cp、改名mv、撤销rm、确定类型file

### 3. 目录操作：建立子目录mkdir、撤销空目录rmdir、改变工作目录cd

### 4. 系统询问：

- 1) 日期时间date（可带参数，用于修改）
- 2) 当前所有用户名、终端名、注册时间who（选项-L显示当前用户数）
- 3) 显示当前目录路径名pwd

### 5. 重定向、管道、通信、后台等

## 三. 重定向与管道命令

### 1. 重定向命令

- 1) Login程序自动设置标准输入为文件为键盘，标准输出文件为屏幕
- 2) \$ cat file1>file2将显示文件file1改为复制到file2
- 3) \$ cat file4>>file2将file4数据添加到file2现有数据后
- 4) \$ wc<file3将统计键入字数改为统计file3字符数
- 5) a.out<file1>file0表示可执行文件a.out重定向为从file1提取数据，结果输出到file0

### 2. 管道命令：建立通道pipe文件，缓冲前一命令的输出，作为后一命令的输入

- 1) 单向性：只从前一命令输入，给后一命令读取
- 2) 同步性：管道满时，前一命令暂停，管道空时，后一命令暂停
- 3) 例：\$ cat file|wc将file的数据给wc命令计数

## 四. 通信命令

### 1. 信箱通信命令mail：非交互非实时式通信

- 1) 以注册名命名私有信箱，以接受者注册名命名信件，存在/usr/spool/mail
- 2) mail命令后的参数为接收者注册名，新行键入信件正文
- 3) .或^D结束输入，读信选项为r：先进先出读信、q：按中断字符（del或return）退出信箱且不改变信箱内容、p：显示全部信件等（不用p选项的话每显示完一个信件会询问一次是否读下一条）

### 2. 对话通信命令write：实时与当前系统其它用户联机通信

- 1) UNIX上允许一个用户或多终端上注册，所以要先用who命令确认
- 2) 再用write user[ttynum]（只注册了一个终端时可缺省终端名）

- 3) 之后对应终端会提示原用户名和终端名
3. 允许或拒绝接受消息的mesg命令
  - 1) n选项表示拒绝写，如正在联机编写重要资料时一般选n
  - 2) y选项表示恢复写许可
  - 3) 不带选项表示只报告当前状态而不改变

## 五. 后台命令

1. 后跟&的命令会被Shell后台执行，其标准输入文件会被定向到/dev/null空文件
2. 可用ps、wait、Kill命令了解和控制后台进程的运行

◆

◆ 联机命令接口的实现

## 一. 键盘终端处理程序

1. 字符接收功能
  - 1) 面向字符：直接将ascii码传给以后程序
  - 2) 面向行：收到行结束符后将行缓冲送给命令解释程序（有时存的是键码，如按下a键后，其键码30存入IO寄存器，由终端处理程序转换成ascii码）
2. 字符缓冲功能
  - 1) 专用缓冲：每个终端各200字符左右的缓冲区，适用于终端较少的情况
  - 2) 公用缓冲：每个缓冲区20字符左右，链成链，数据传给程序后重新链入空缓冲区链，提高缓冲利用率
3. 回送显示：硬件实现较快但不灵活；软件实现方便大小写切换和密码输入
4. 屏幕编辑：删除字符（移出字符队列末字符、如backspace或ctrl h）、删除行、插入、移动光标、屏幕上卷或下移等
5. 特殊字符处理
  - 1) 中断字符：如break、delete、ctrl c，中断处理程序将发出软中断信号，各进程收到后自我终止
  - 2) 停止上卷、恢复上卷：如ctrl s、ctrl q并不会被存储，而是用于设置中断数据结构的上卷标志，终端视图输出新数据前需检查该标志

## 二. MS-DOS解释程序

- 1) 命令解释程序一般放在用户层，以用户态运行
- 2) MS-DOS是微软1981开发的微机OS，其命令解释程序为COMMAND.COM
1. 命令解释程序的作用：提示用户键入命令，并读入、识别、处理命令，转交控制权给对应处理程序
2. 命令解释程序的组成
  - 1) 常驻部分：正常退出中断INT20、驻留退出中断INT27、错误退出中断INT24；检查程序终止后是否被覆盖、若被覆盖则重新调入内存
  - 2) 初始化部分：每次系统接电或重启后找到AUTOEXEC.BAT文件并装入基址执行，因为只需初始化一次，第二个装入的文件就会覆盖它
  - 3) 暂存部分：命令解释程序等，可被用户程序覆盖，运行完后再从磁盘调回

命令解释程序的工作流程：系统接电后控制权先



- 交给初始化部分，  
AUTOEXEC.BAT执行完后
3. 转交控制权给暂存部分，  
读入键盘缓冲命令，判断  
出对应内部命令并转交控  
制权或通过系统调用exec  
装入该外部命令，再转交

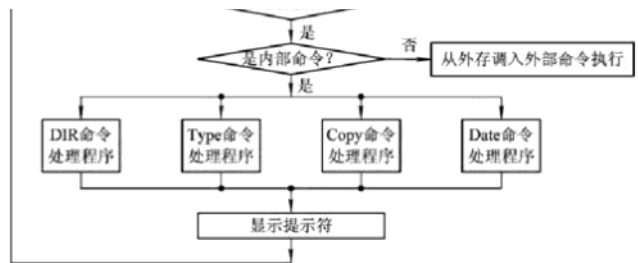


图9-2 COMMAND.COM的工作流程

### 三. Shell解释程序

- 1) Shell是Linux或UNIX的系统最外层，所有命令都是经过Shell的解释/识别再传给内核的程序处理的
1. Shell命令的特点
  - 1) 一条命令行可以有多个命令，解释后可能要产生多个命令处理进程
  - 2) 命令间可能有不同分隔符，如:顺序执行、&后台执行、|管道执行
2. 二叉树结构的命令行树
  - 1) 命令表型结点: ;和&分隔符作为结点，左右部分构成其左右子树
    - (1) ;结点是递归执行完左子树才能执行右子树
    - (2) &结点是启动左子树后即可执行右子树
  - 2) 管道文件型结点: |分隔符作为结点，左右部分构成其左右子树
  - 3) 简单命令型结点: 是内部命令就直接执行，否则建个新子进程，运行完毕后再恢复Shell执行
3. Linux命令解释程序的工作流程
  - (1) 内核为每个终端用户建立一个进程执行Shell解释程序
    - 1) 读取键入的命令行
    - 2) 分析命令、建立二叉树、以命令名为文件名、将参数改造为系统调用execve内部处理时要求的形式
    - 3) 终端进程调用fork为每一命令建立子进程
    - 4) 等待子进程完成。;结点是通过系统调用Wait4(), 调用execve()查找对应程序调入内存执行，结束后唤醒父进程执行下一条；&结点启动左子树后可立刻执行右子树

5)

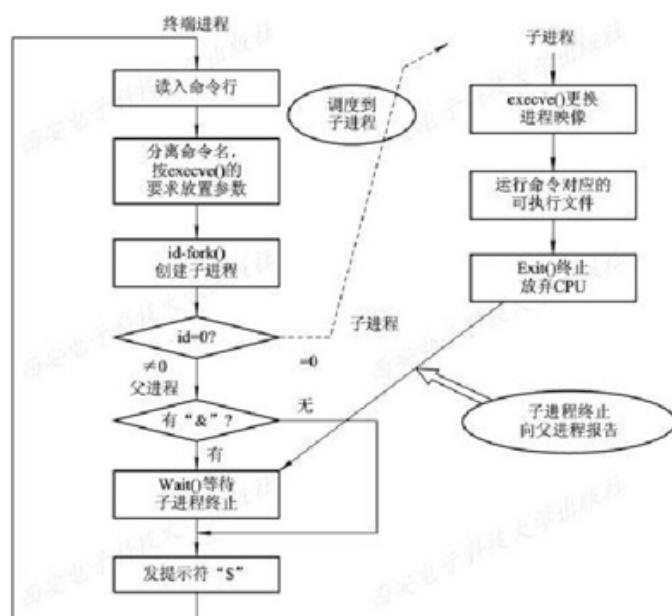


图9-5 Shell基本执行过程及父子进程之间的关系

- i.
- ii.
- iii.
- iv.
- v.
- vi.
- vii.
- viii. -----我是底线-----